This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

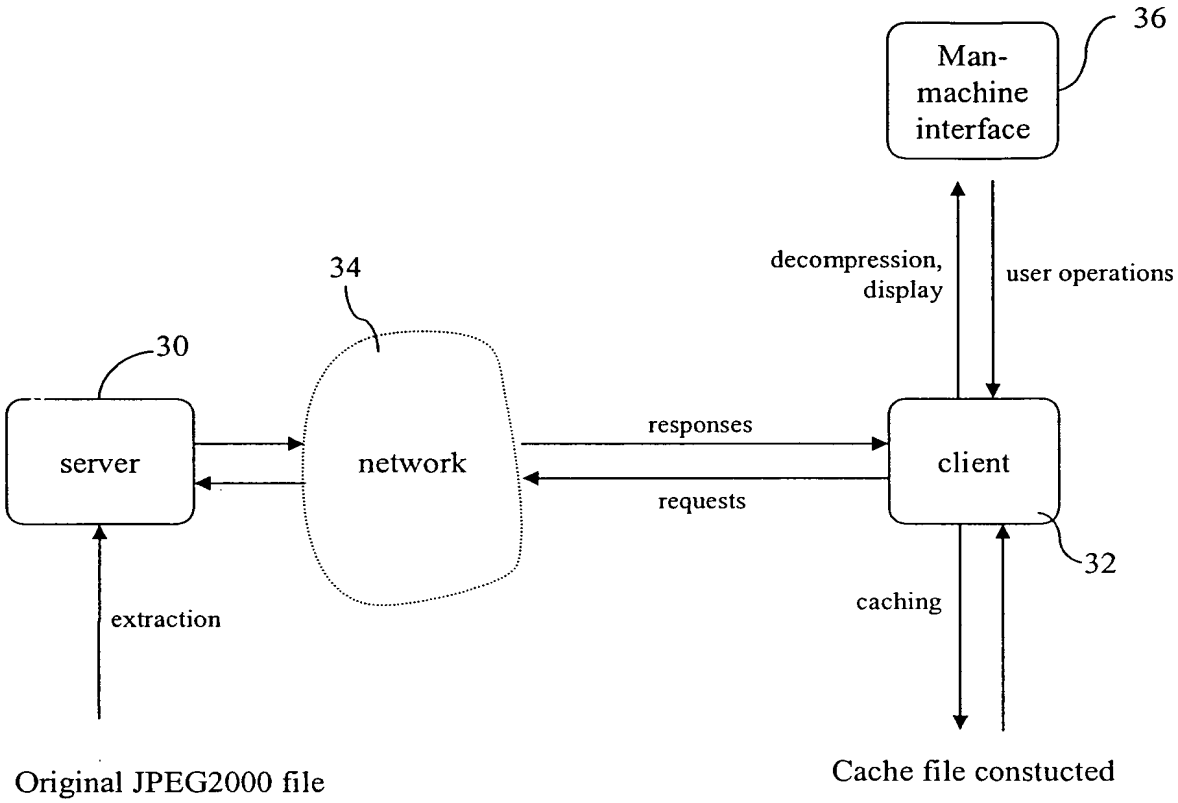
Man-
machine
interface
36
decompression,
display
user operations
34
30
server
network
responses
requests
client
32
extraction
caching
Original JPEG2000 file
Cache file constucted

Figure 1

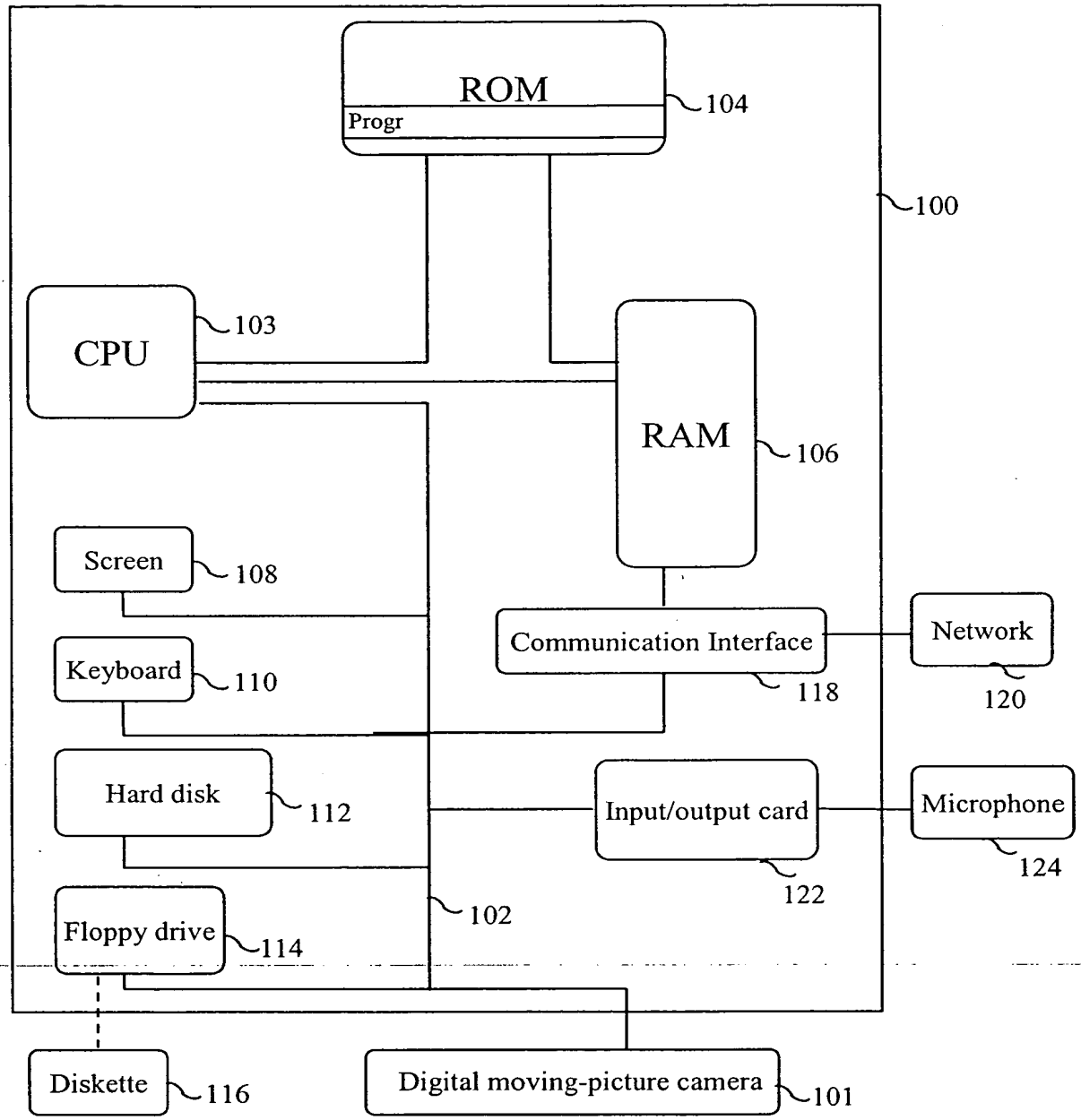
ROM
Progr
104
100
CPU
103
RAM
106
Screen
108
Keyboard
110
Communication Interface
118
Network
120
Hard disk
112
Input/output card
122
Microphone
124
102
Floppy drive
114
Diskette
116
Digital moving-picture camera
101

Figure 2

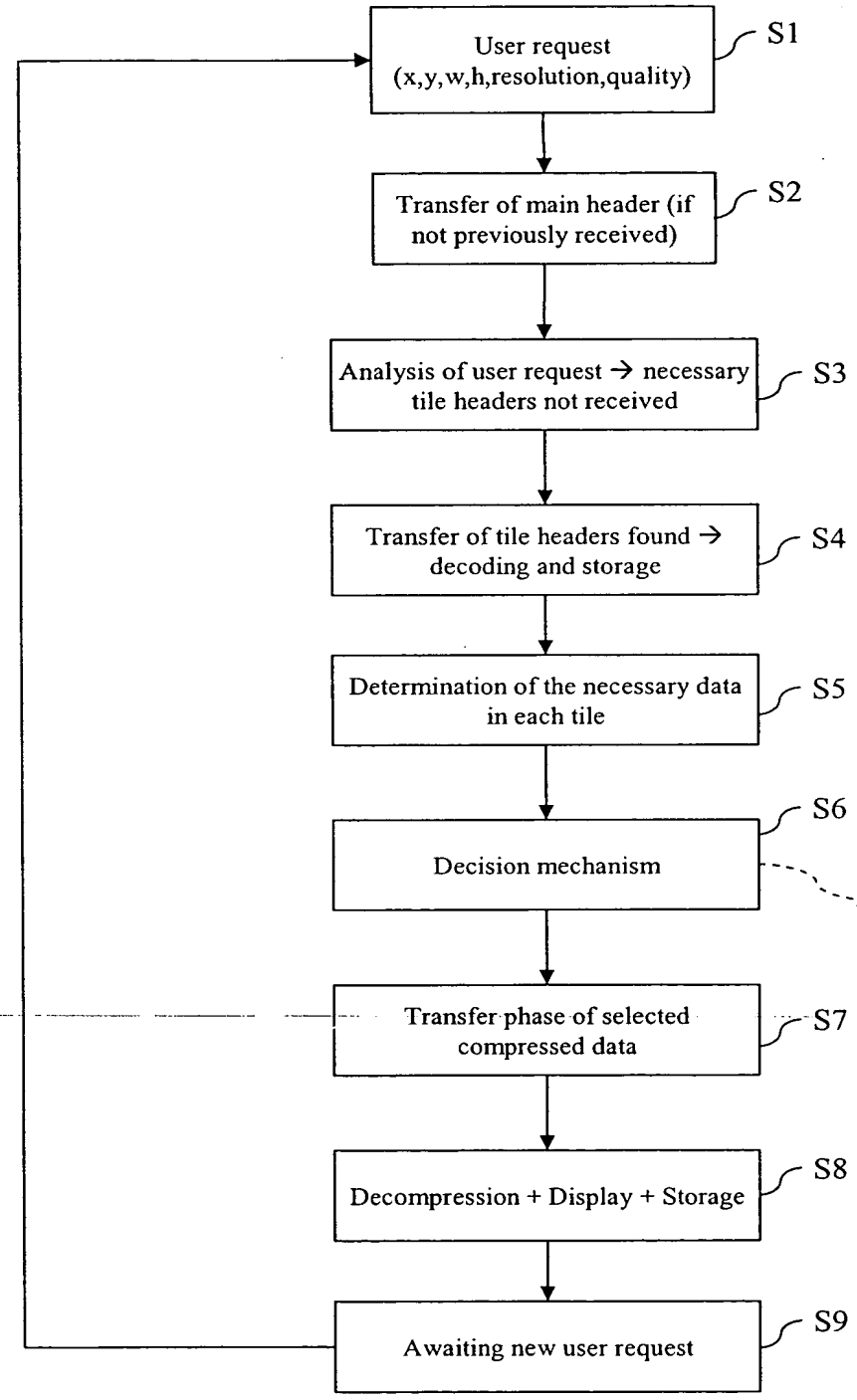
User request
(x,y,w,h,resolution,quality)
S1
Transfer of main header (if not previously received)
S2
Analysis of user request → necessary tile headers not received
S3
Transfer of tile headers found → decoding and storage
S4
Determination of the necessary data in each tile
S5
Decision mechanism
S6
Figures 4, 7
Transfer phase of selected compressed data
S7
Decompression + Display + Storage
S8
Awaiting new user request
S9

Figure 3

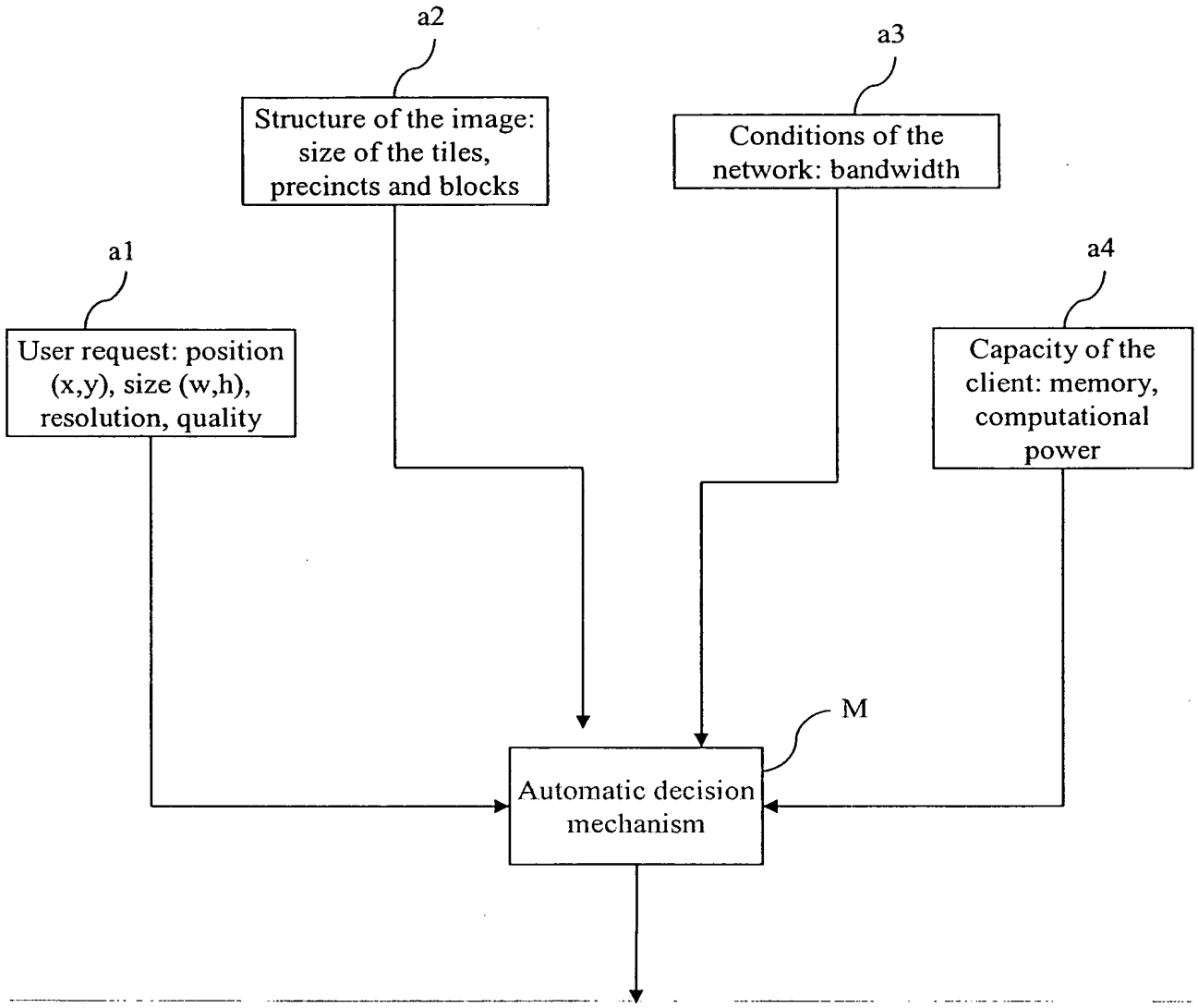
a2
Structure of the image: size of the tiles, precincts and blocks
a3
Conditions of the network: bandwidth
a1
User request: position (x,y), size (w,h), resolution, quality
a4
Capacity of the client: memory, computational power
M
Automatic decision mechanism
Type of JPEG2000 data sent back:
set of tiles and/or precincts and/or blocks

Figure 4

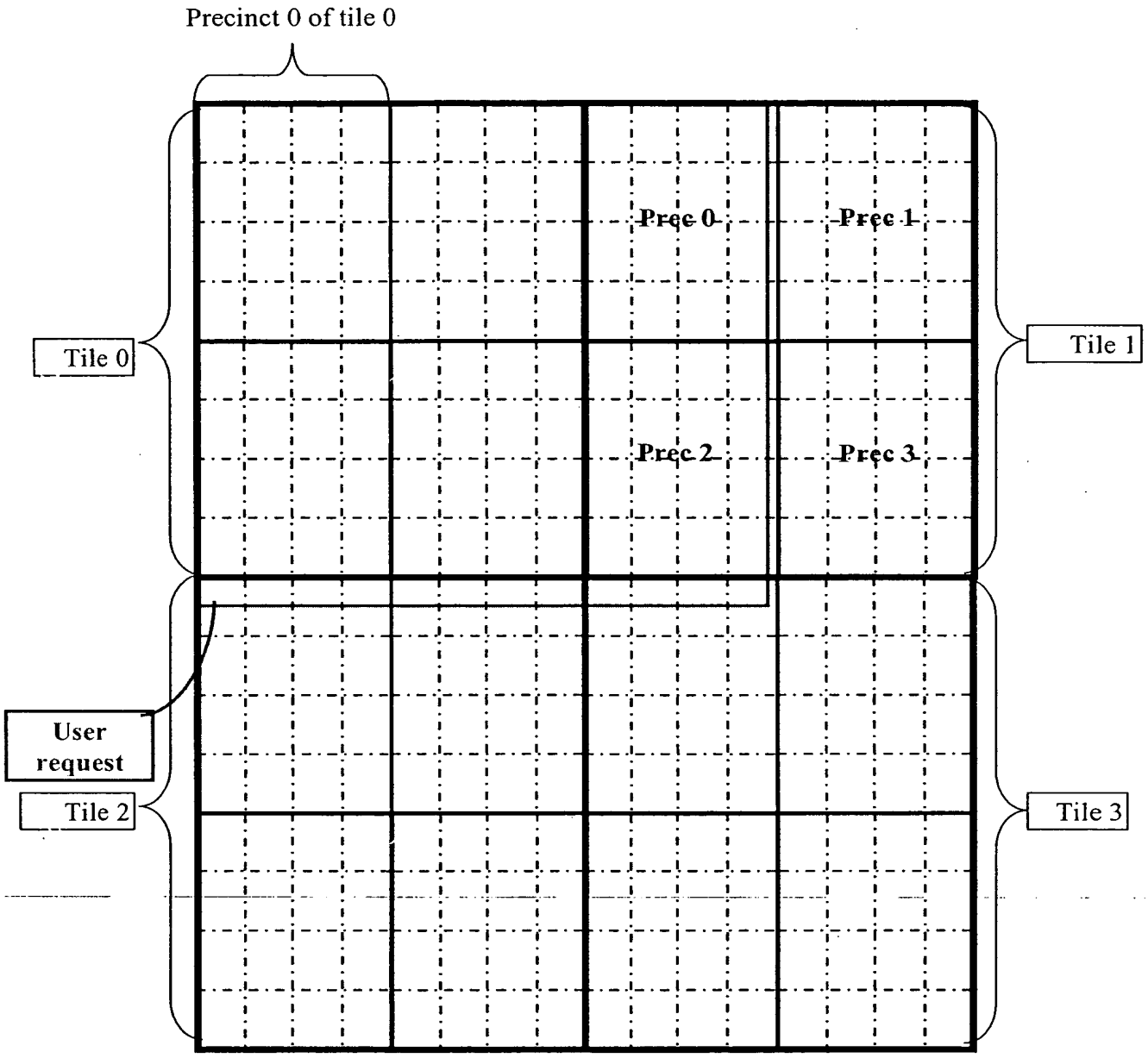
Precinct 0 of tile 0
Prec 0
Prec 1
Prec 2
Prec 3
Tile 0
Tile 1
Tile 2
Tile 3
User request

Figure 5

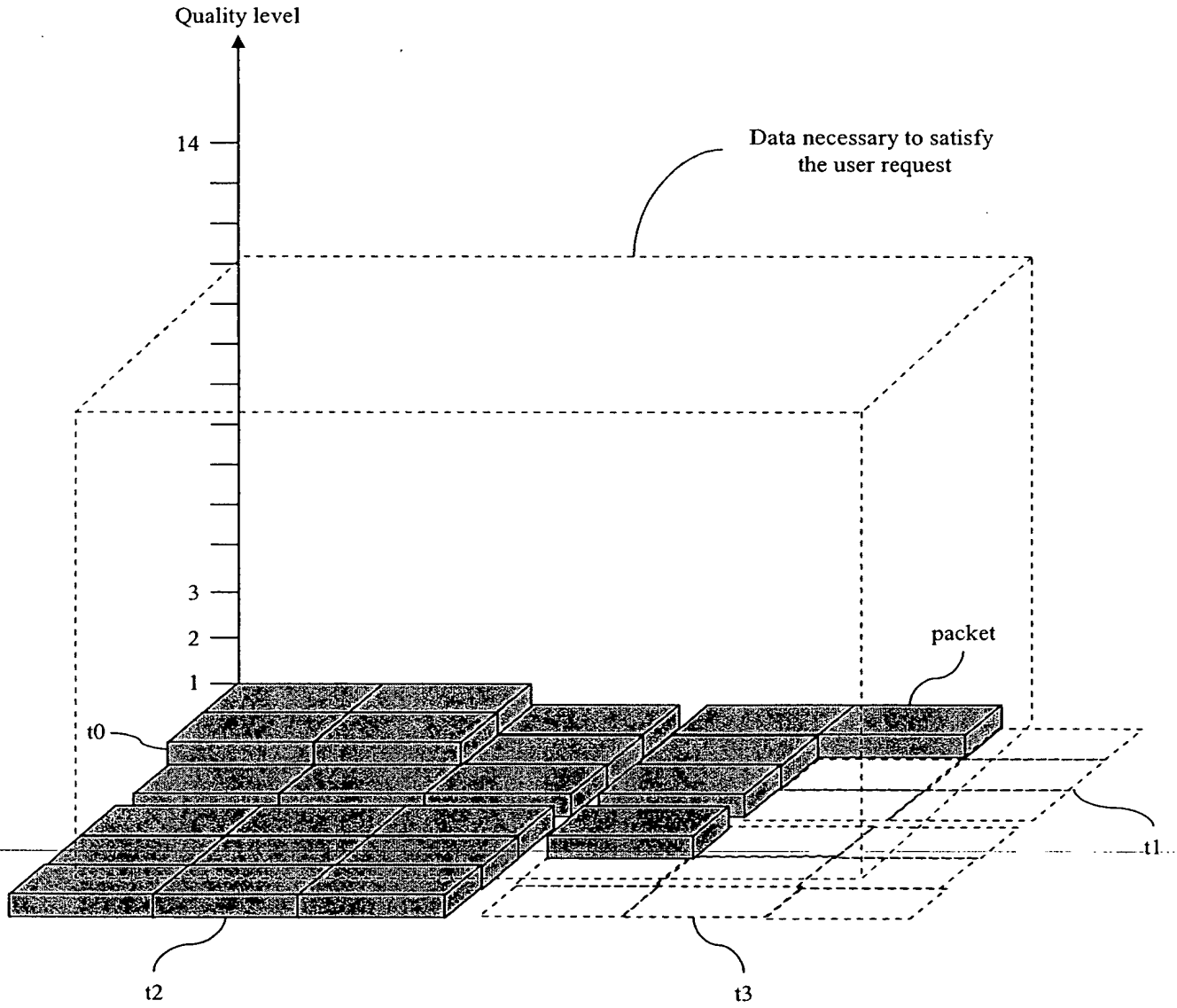
Quality level
14
3
2
1
Data necessary to satisfy
the user request
packet
t0
t1
t2
t3

Figure 6

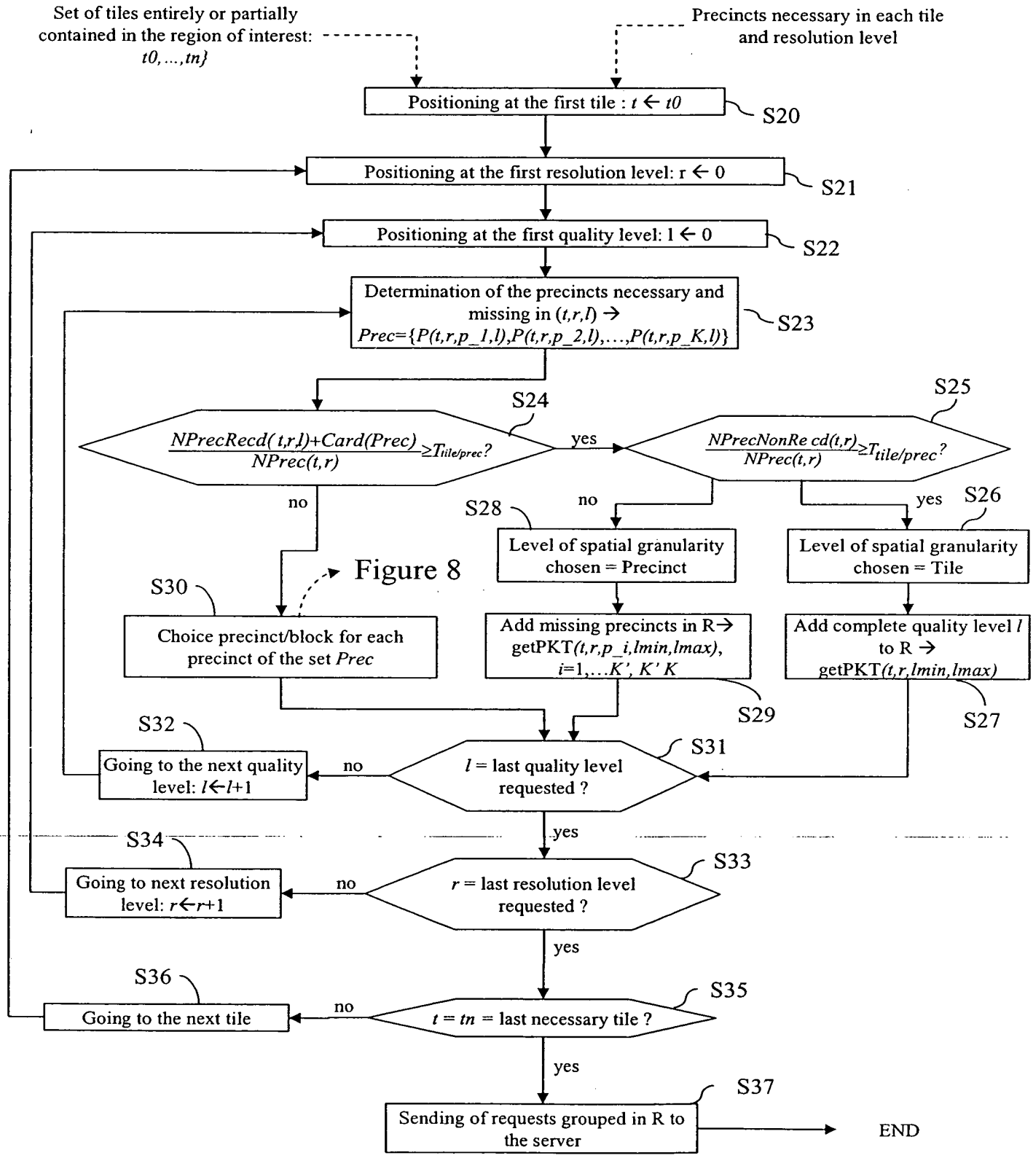

Figure 8

Figure 7

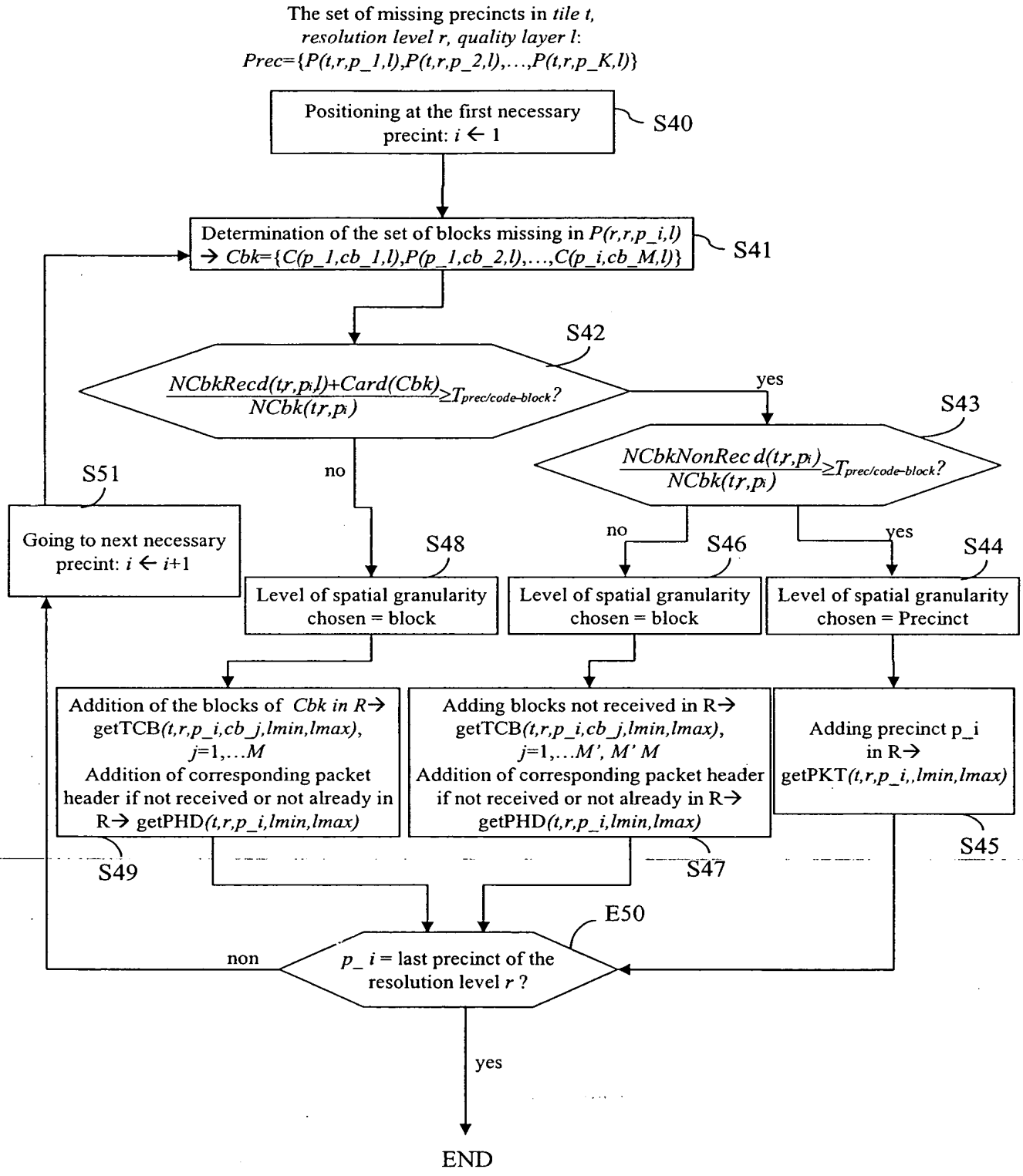
The set of missing precincts in tile t, resolution level r, quality layer l: Prec={P(t,r,p_1,l),P(t,r,p_2,l),...,P(t,r,p_K,l)}
Positioning at the first necessary precint: i ← 1
S40
Determination of the set of blocks missing in P(r,r,p_i,l) → Cbk={C(p_1,cb_1,l),P(p_1,cb_2,l),...,C(p_i,cb_M,l)}
S41
S42
NCbkRecd(t,r,pi,l)+Card(Cbk) / NCbk(t,r,pi) ≥ Tprec/code-block?
yes
S43
NCbkNonRecd(t,r,pi) / NCbk(t,r,pi) ≥ Tprec/code-block?
no
yes
S51
Going to next necessary precint: i ← i+1
S48
Level of spatial granularity chosen = block
no
S46
Level of spatial granularity chosen = block
yes
S44
Level of spatial granularity chosen = Precinct
Addition of the blocks of Cbk in R→ getTCB(t,r,p_i,cb_j,lmin,lmax), j=1,...M Addition of corresponding packet header if not received or not already in R→ getPHD(t,r,p_i,lmin,lmax)
S49
Adding blocks not received in R→ getTCB(t,r,p_i,cb_j,lmin,lmax), j=1,...M', M' M Addition of corresponding packet header if not received or not already in R→ getPHD(t,r,p_i,lmin,lmax)
S47
Adding precinct p_i in R→ getPKT(t,r,p_i,,lmin,lmax)
S45
E50
non
p_ i = last precinct of the resolution level r ?
yes
END

Figure 8

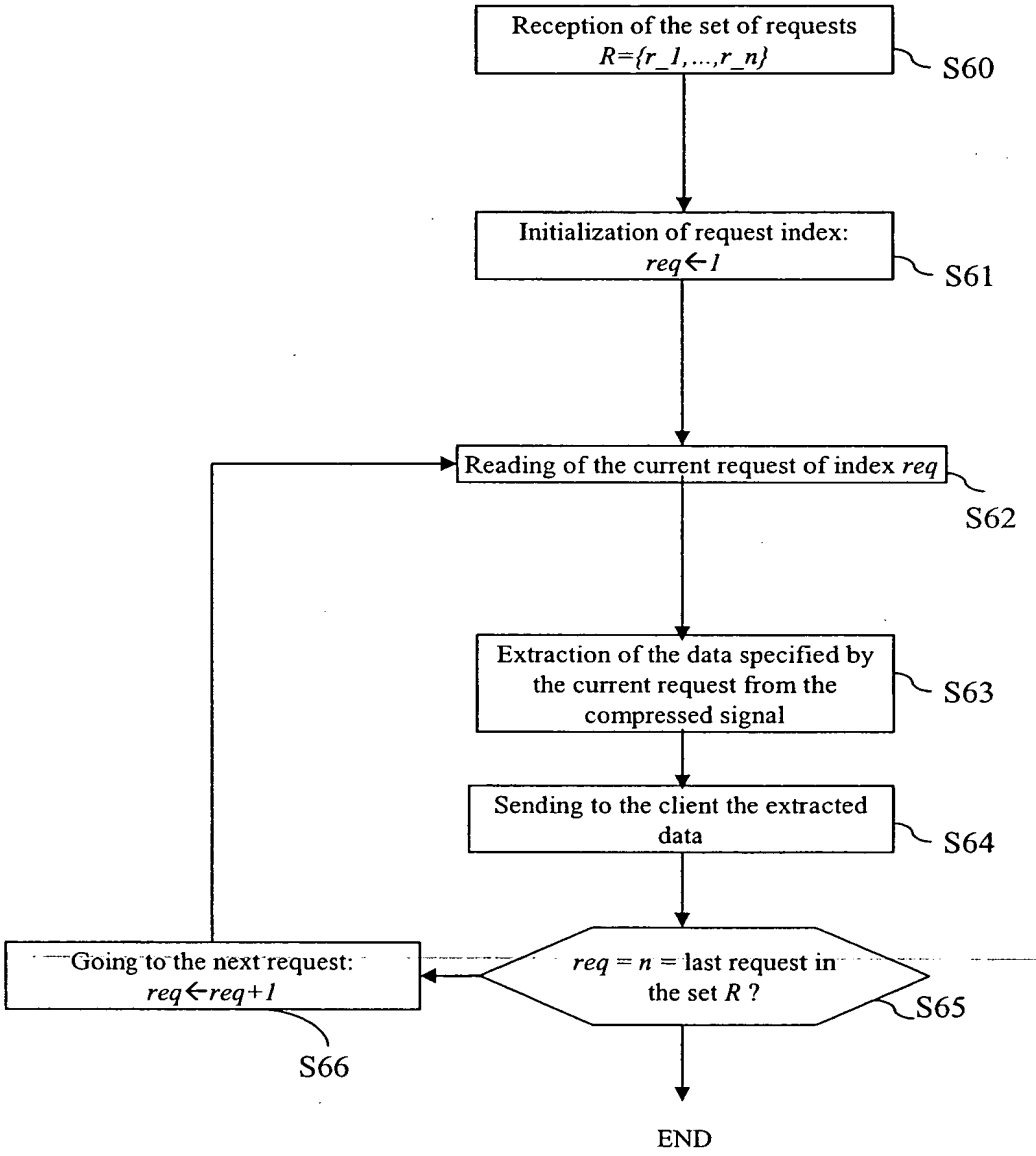
Reception of the set of requests
R={r_1,...,r_n}
S60
Initialization of request index:
req←1
S61
Reading of the current request of index req
S62
Extraction of the data specified by the current request from the compressed signal
S63
Sending to the client the extracted data
S64
req = n = last request in the set R ?
S65
Going to the next request:
req←req+1
S66
END

Figure 9

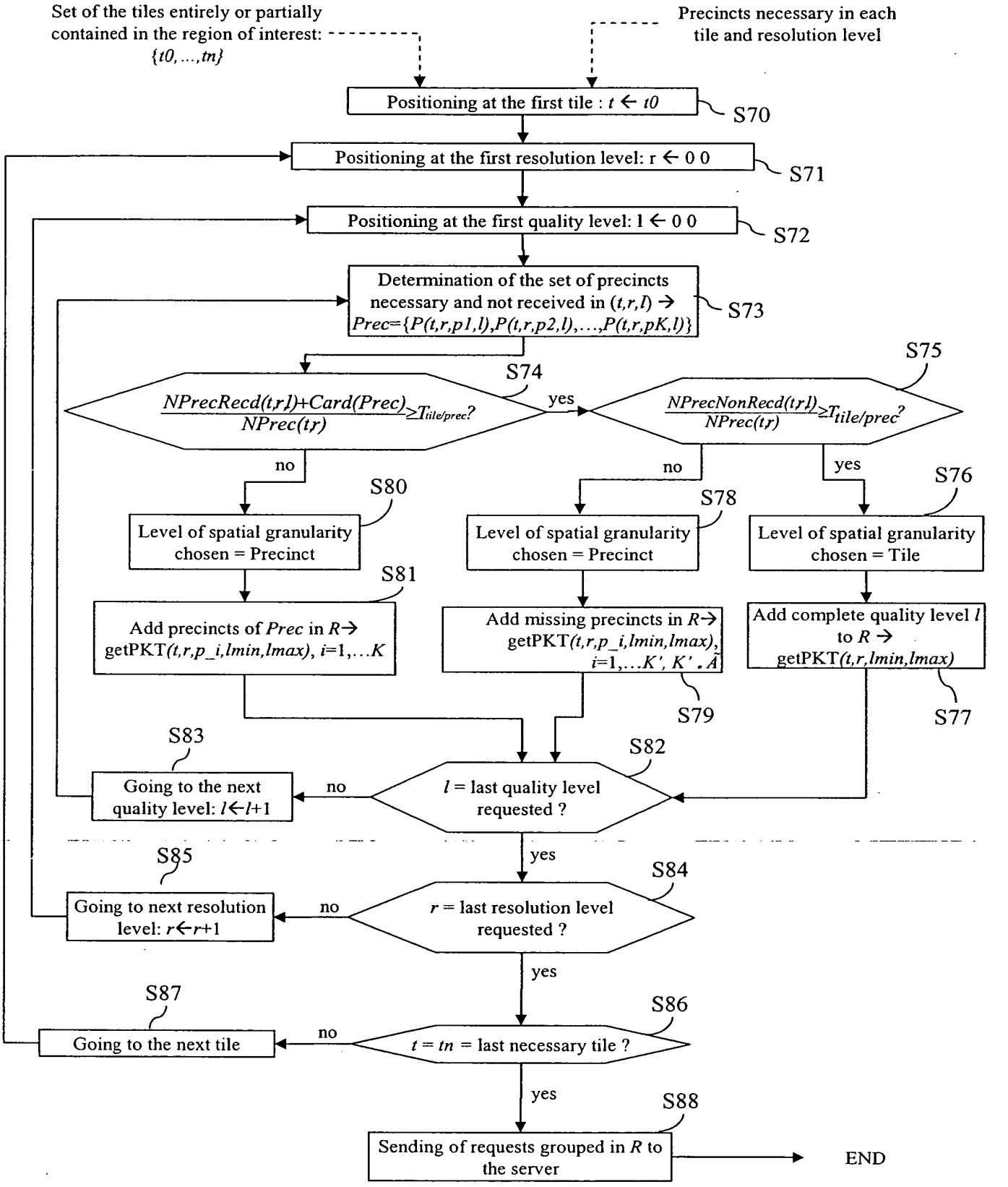
Set of the tiles entirely or partially contained in the region of interest: {t0,...,tn}
Precincts necessary in each tile and resolution level
Positioning at the first tile : t ← t0
S70
Positioning at the first resolution level: r ← 0 0
S71
Positioning at the first quality level: l ← 0 0
S72
Determination of the set of precincts necessary and not received in (t,r,l) → Prec={P(t,r,p1,l),P(t,r,p2,l),...,P(t,r,pK,l)}
S73
S74
S75
yes
no
yes
no
yes
S80
S78
S76
Level of spatial granularity chosen = Precinct
Level of spatial granularity chosen = Precinct
Level of spatial granularity chosen = Tile
S81
Add precincts of Prec in R→ getPKT(t,r,p_i,lmin,lmax), i=1,...K
Add missing precincts in R→ getPKT(t,r,p_i,lmin,lmax), i=1,...K', K' < K
Add complete quality level l to R → getPKT(t,r,lmin,lmax)
S79
S77
S83
S82
Going to the next quality level: l←l+1
no
l = last quality level requested ?
yes
S85
S84
Going to next resolution level: r←r+1
no
r = last resolution level requested ?
yes
S87
S86
Going to the next tile
no
t = tn = last necessary tile ?
yes
S88
Sending of requests grouped in R to the server
END

Figure 10

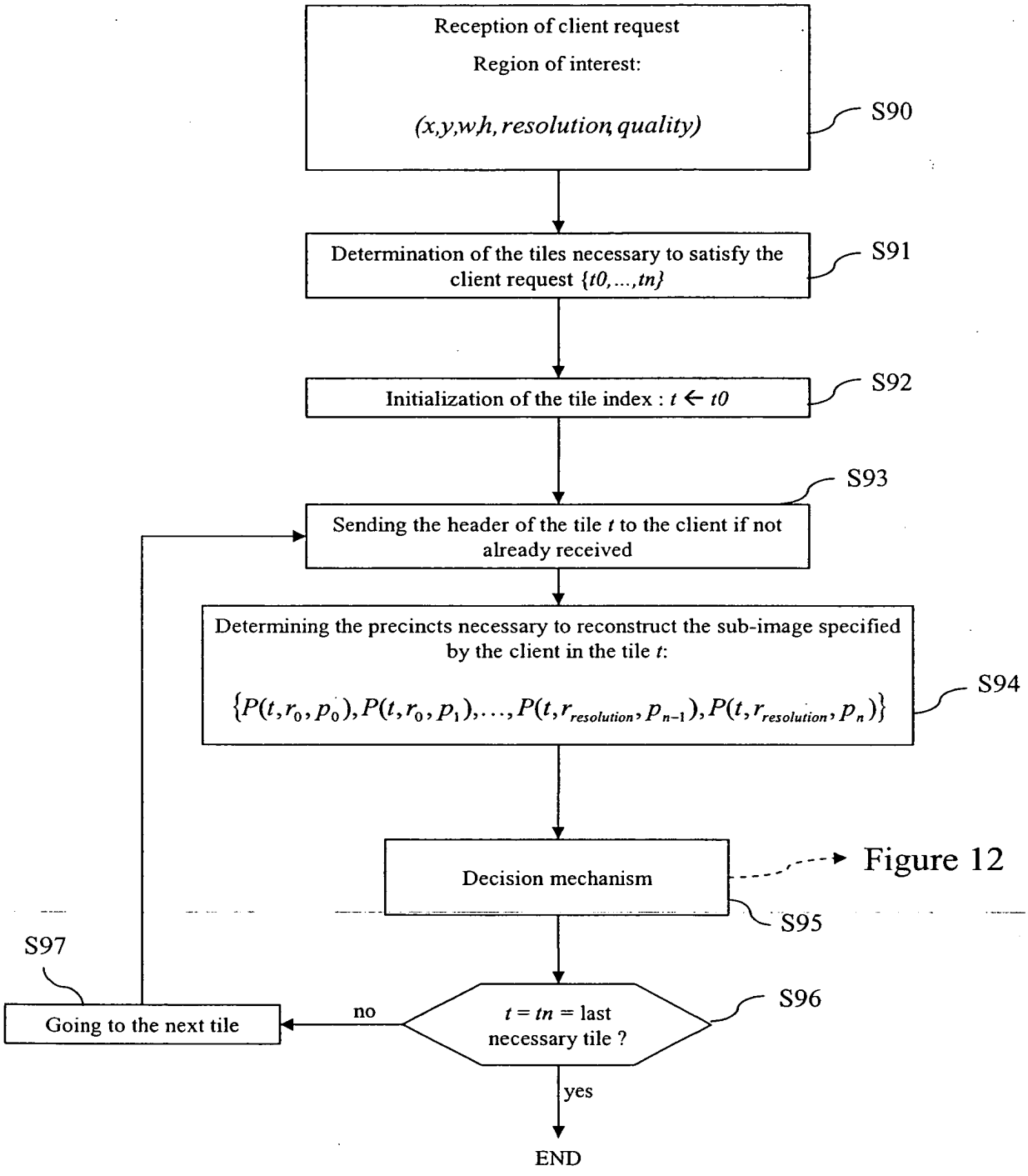

Figure 12

Figure 11

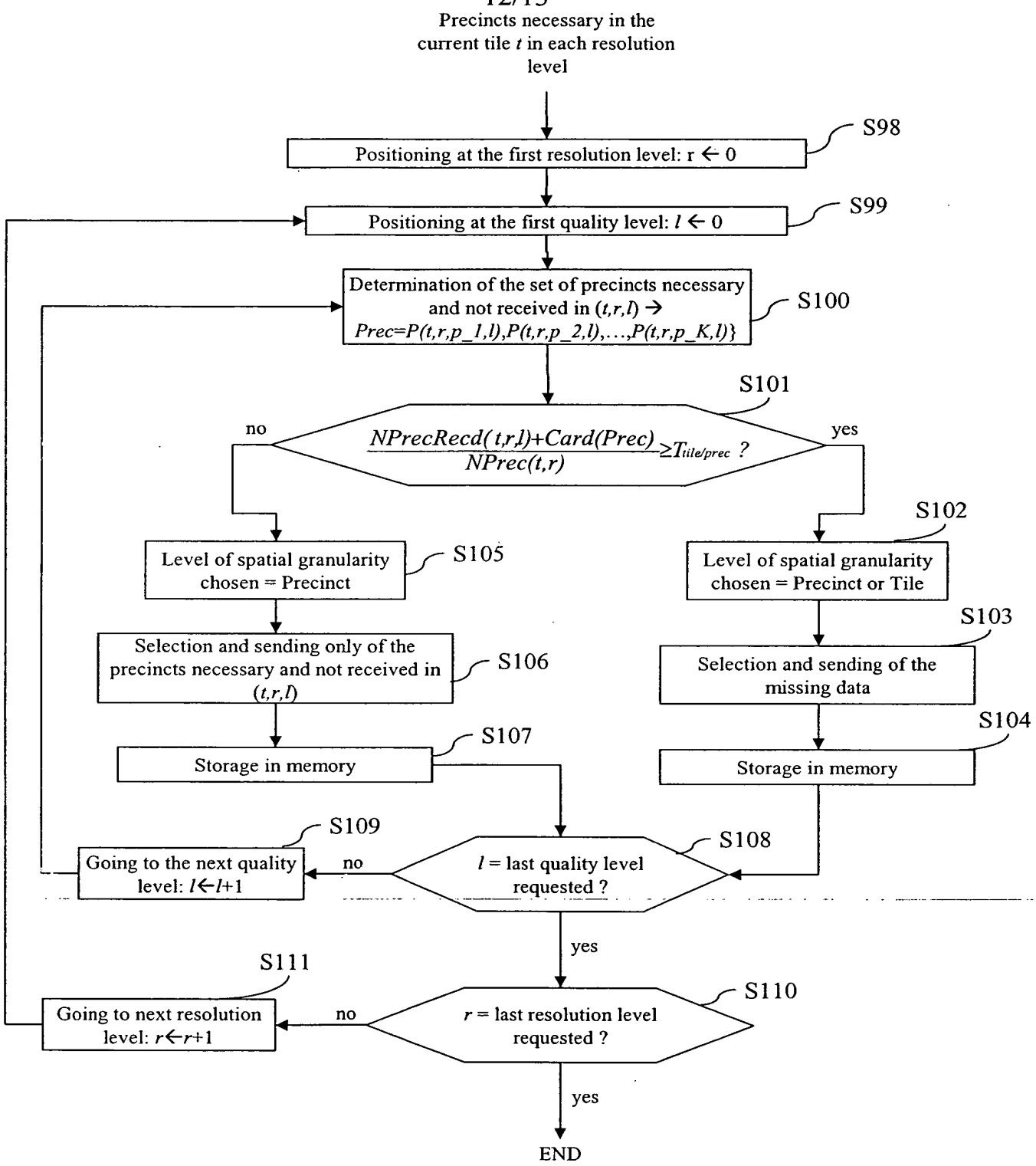
Precincts necessary in the current tile t in each resolution level
Positioning at the first resolution level: r ← 0
S98
Positioning at the first quality level: l ← 0
S99
Determination of the set of precincts necessary and not received in (t,r,l) → Prec=P(t,r,p_1,l),P(t,r,p_2,l),...,P(t,r,p_K,l)}
S100
S101
no
NPrecRecd(t,r,l)+Card(Prec) / NPrec(t,r) ≥Ttile/prec ?
yes
S102
Level of spatial granularity chosen = Precinct
S105
Level of spatial granularity chosen = Precinct or Tile
S103
Selection and sending only of the precincts necessary and not received in (t,r,l)
S106
Selection and sending of the missing data
S104
Storage in memory
S107
Storage in memory
S109
S108
Going to the next quality level: l←l+1
no
l = last quality level requested ?
yes
S111
S110
Going to next resolution level: r←r+1
no
r = last resolution level requested ?
yes
END

Figure 12

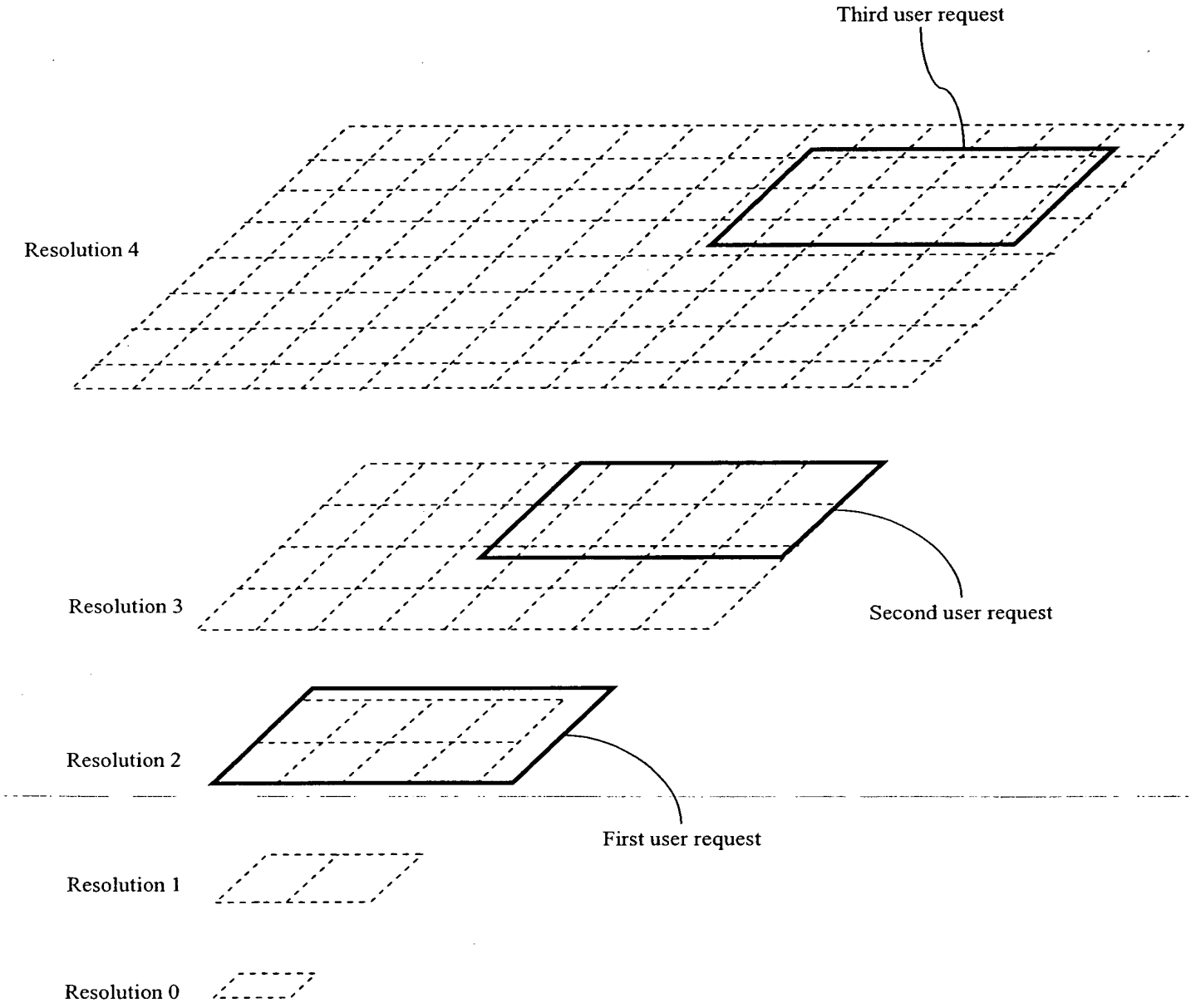
Third user request
Resolution 4
Resolution 3
Second user request
Resolution 2
First user request
Resolution 1
Resolution 0

Fig. 13